Towards Principled Experimental Study of Autonomous Mobile Robots

(Appears in ISER95. To appear in *Autonomous Robots*.)

Erann Gat
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
gat@robotics.jpl.nasa.gov

ABSTRACT

We review the current state of research in autonomous mobile robots and conclude that there is an inadequate basis for predicting the reliability and behavior of robots operating in unengineered environments. We present a new approach to the study of autonomous mobile robot performance based on formal statistical analysis of independently reproducible experiments conducted on real robots. Simulators serve as models rather than experimental surrogates. We demonstrate three new results: 1) Two commonly used performance metrics (time and distance) are not as well correlated as is often tacitly assumed. 2) The probability distributions of these performance metrics are exponential rather than normal, and 3) a modular, object-oriented simulation accurately predicts the behavior of the real robot in a statistically significant manner.

1. Introduction

There appears to be an unfortunate dichotomy in autonomous-mobile-robotics research between theory and practice. Published reports in this particular area of robotics seem to fall largely into one of two categories: theoretical work with little or no experimental verification (except, on occasion, in simulation), and anecdotal experimental results from implemented systems with little or no formal theoretical foundation. It is rare to find a formal theoretical prediction verified (or refuted) by independently reproducible experiments performed on a real robot. It is even rarer to find such results supported by an analysis of their statistical significance. Control experiments are nearly unheard of.

For example, in a cursory survey of 44 papers in the mobile robotics track of the 1994 International Conference on Robotics and Engineering (track 5, excluding four papers on legged robots) we found seventeen papers that describe work done on an actual mobile robot [1-3,5,7,8,10-14,16,20-23,25]. Of these, only three reported quantitative results from more than one experimental trial [2,3,5]. (A few papers claim to have produced such results but do not actually report them, e.g. [22].) Of these three, only one [5] deals directly with autonomous control. While such a shallow survey of the literature does not prove anything, it is indicative that a problem exists.

This is not necessarily an indictment of the research community. I have argued elsewhere that this dichotomy between theory and practice in the study of autonomous mobile robots is due to an inherent and unavoidable property of the problem: autonomous mobile robots must interact with complex environments which have not been engineered specifically for the robot. Interactions with such environments are extremely difficult to model because they are governed by an enormous number of independent variables [4].

In order to make the mathematics tractable, standard analytical approaches often assume that most of these independent variables can be safely ignored (or at least that their consideration can be deferred). For example, there is a vast theoretical literature on the path-planning problem, which is almost invariably posed as a purely geometrical problem where the quality of a solution is measured in terms of path length (e.g. Latombe [15]). Such formulations routinely ignore such factors as computational costs, sensor noise, occlusions and resolution limits, and mechanical interactions between a robot and a supporting surface, including friction and surface deformation.

Likewise, in order to make experimentation tractable, issues such as controlling for extraneous effects and statistical significance of results are routinely ignored. It is rare to find a description of an experimental setup that is sufficiently detailed to allow the experiment to be independently reproduced. The choice of the number of experiments to conduct is usually made on a purely *ad hoc* basis (reporting the result of a single experimental trial is common), and control experiments and statistical analysis are all but nonexistent. (To quote Matthew Ginsberg, I myself am hardly innocent in this regard.) As a result there is a lot of passionate debate, but no objective basis for evaluating the relative merits of different approaches to the problem of autonomous control.

We can no longer afford to sweep these issues under the rug. In 1996 NASA will launch an autonomous mobile robot to explore the surface of Mars. This robot represents a substantial expenditure of taxpayer money, and so it is important to accurately assess the reliability of our control methodology before launch. Furthermore, the harsh realities of the Martian environment do not permit us the luxury of making arbitrary simplifying assumptions in our theories, even if those assumptions appear intuitively plausible. Reality, not theorems, is our ultimate arbiter of truth.

2. Approach

Our approach is to treat the experimental study of mobile robots in the manner of a natural science or an empirical engineering discipline. The natural sciences (e.g. biology) regularly study the interactions of systems that are as complex or more than the environments mobile robots interact with. It is usually impossible to model such complex systems starting from first principles. Instead, probability theory is used to model system components as random processes. Experiments are designed to measure sampling distributions of the resulting random variables, and statistical methods are used to analyze the results.

One important result of our work is that two commonly used performance metrics turn out to have probability distributions that appear to be exponential rather than normal. Most of the standard techniques used in the natural sciences assume normal distributions. We will therefore be forced to rely on some non-traditional analysis methods, known as non-parametric methods, which do not rely on the probability distribution having any particular shape.

Although we will emphasize experimental results (we hope to describe our apparatus, methods, and models in sufficient detail to allow independent replication), we will also construct models of our robot systems. These models will take the form of simulations. The measure of our simulations, however, will be how well they predict real experimental results, not whether they reflect a particular mathematical formalism. This is an important distinction. In the mobile robot literature simulators are commonly used as experimental testbeds to "demonstrate" the validity of mathematical models derived from first principles. In our case, the simulator is the model, and the validation comes from experiments on a real robot.

We also wish to highlight the *process* leading to our results, and to contrast that process with current practice. In our view, the current practice places emphasis on *getting things to work*, resulting in the widespread use of iterative design interleaved with *ad hoc* evaluation. This approach often results in working systems, but it does not yield an understanding of the limitations of these systems. In particular, it provides little assurance that a system will continue to operate when environmental parameters are changed. This is of particular concern to NASA because we cannot test our robot in the actual conditions under which it will expected to operate (since, among other reasons, we do not know exactly what those conditions will be).

Our approach therefore emphasizes *reliable prediction of system performance*. We intend our approach to complement the current practice rather than supplant it. Ultimately our goal is to build systems that work. Reliable measurement of system performance is a necessary component of the process, but cannot replace the current iterative design methodologies, which we continue to advocate.

3. Apparatus

The robot under study is Rocky 3.2 [18], a rebuilt version of Rocky 3 [6]. This robot has essentially the same chassis design, size, computer and sensor suite as those designed for the actual flight rover, which is known as the MFEX (Mars Flight EXperiment) rover. (See figure 1.) Both rovers are six-wheel rocker-bogie type vehicles. Each has an 8085 processor with 1/4 megabyte of bank-switched RAM, most of which is used to hold image data. The rover is programmed in C and 8085 assembly language. The main differences between the two vehicles are that Rocky 3.2 weighs 20 kg, about twice as much as the flight rover, uses rubber instead of metal tires, and lacks some of the contact and motor current sensors planned for the flight rover.

The primary sensor on the robot is a structured-light range sensor, comprising five laser diodes each equipped with a cylindrical lens, and a pair of stereo cameras. The camera data is clocked out and captured in software by the 8085, using a scheme similar to that used by Horswill and Yamamoto [9]. The data is processed to produce a 5x4 array of range measurements, which undergo a coordinate transformation to become a 5x4 array of height measurements. Adjacent height measurements are compared, and if the difference exceeds a threshold, that area is assumed to be untraversable. The effectiveness of this sensor and data processing scheme is under separate study. The rover also has articulation sensors and inclinometers, but these were not used in the work described here.

The rover also has wheel encoders and a rate gyroscope, which it uses to keep track of its position through dead reckoning. The rate gyroscope tends to drift, and the wheels slip in loose soil, resulting in dead reckoning errors. A preliminary analysis of these errors appears in [18].

All experiments were performed indoors in a 4 m by 12 m sandbox filled with loose sand and crushed red brick to simulate the reddish color of Martian soil. (Using a better Martian soil simulant is impractical because Mars soil contains sub-micron dust, which can be toxic to humans if inhaled.) The sandbox was instrumented with a tracking system comprising four overhead CCD cameras, whose field of view covered 10 m of the sandbox's length. The rover was equipped with a visual target that could be easily identified and located in the images. The system was calibrated by comparing the readings of the tracking system to ground-truth measurements obtained by placing the target at reference locations. The reference locations were marked by an array of strings stretched between nails placed in the wooden side rails of the sandbox at 1 m intervals. The results showed that the system is accurate to within about 1 cm and 1 degree. The overhead tracking system was used only for gathering experimental data, and was not accessible to the rover navigation software.

The sandbox was made as large as the available space would allow. Unfortunately, this turned out not be large enough. Initial experiments revealed that the rover would often approach the edges of the sandbox when avoiding obstacles. The rover would then detect the sandbox side rails as obstacles, which would affect the experimental results in unrealistic ways (since long, straight obstacles like the side rails are unlikely to exist in Mars).

To address this problem we implemented a "virtual sandbox" in the experiment management software. During a run if the rover came within 90 cm of an edge (as measured by the overhead tracking system) the experiment manager program would command the rover to stop and turn in place so that its heading was reflected about an axis parallel to the edge of the sandbox. The goal location was also reflected about this same axis. The net effect was to create a "virtual sandbox" adjacent to and a mirror image of the original sandbox. This reflection could be repeated to produce a sandbox of effectively infinite size, but in practice only one reflection on either side of the physical sandbox has been used to date.

This technique allows runs of unlimited length in a sandbox of finite size. The supervisory program keeps track of the rover's "virtual position", and issues the proper commands whenever the rover nears an edge. The main limitation of the method is that the terrain in the "infinite" sandbox is just repeated mirror images of the terrain in the original sandbox, and so the obstacle distributions in the virtual sandbox are not quite random. Nevertheless, this is a useful technique for gathering data in an experimental area which is not quite big enough by itself for realistic tests.

4. Method

In our experiments the quantities that we measure are the values of two performance metrics: traverse distance and elapsed time to reach a goal. These are commonly used metrics, but in fact the choice of these metrics is based more on convenience than on sound theoretical considerations. Time and distance are easy to measure. There are many other performance metrics we could have chosen, many of which affect the outcome of a mission much more directly than time or distance (energy consumption, for example). Many papers on optimal path planning focus on a single performance metric, usually time or distance, presumably on the tacit assumption that other performance metrics are more or less correlated, and that if one optimizes, say, path length then traverse time, energy consumption, etc. will also be optimized. We will show that this assumption may not be valid. The present experiments could be improved by measuring energy consumption, but we currently do not have the means to do so.

To conduct an experiment we first generate a test course by placing rocks of various sizes in the sandbox. The size and placement of these rocks is chosen according to a published model of the rock distributions on Mars, to be described shortly. We then instruct the rover to travel from one end of the sandbox to the other, between two predetermined locations that are 7.6 meters apart. (This is the longest traverse that is possible within the constraints imposed by the field of view of the overhead cameras.) The route that the rover takes is recorded by tracking the rover using the overhead cameras. The rover's position and orientation are recorded at regular intervals, along with a time stamp. Occasionally the overhead tracking system will lose track of the rover and cause a delay while the rover is reacquired. The time stamps are corrected to account for this delay.

A test course is constructed by first choosing an obstacle density. This density can be chosen arbitrarily (for example, to study the effects of gradually increasing obstacle densities on the performance of a particular navigation algorithm), or it can be chosen according to the standard model of Martian terrain, developed by Moore [19]. According to Moore's model, the number of rocks per square meter with a diameter less than or equal to a given size D is:

$$N=kD^{-2.66} \qquad (1)$$

where k is a parameter that varies according to the particular location on the surface. This model is an empirical fit to the rock size distributions observed at the two Viking lander sites. It is accurate only for D > 14 cm, which is fortuitous since this is about the smallest size rock that the laser ranging sensor will detect as an obstacle. If one assumes that the Moore model distribution holds over the entire planet, then thermal inertia data indicate that the modal value of k for all of Mars is approximately 0.00415; this case is referred to as "Mars-nominal" terrain.

Locations for obstacles are generated using a random number generator. Rocks are placed at the prescribed location using a tape measure. Setting up a test course is time-consuming work, so each course was used for four runs, two in each direction. This also allows some interesting statistical analysis to be done to determine, for example, how repeatable (and thus how predictable) the rover's performance is in a given terrain.

The rover moves in discrete steps, where each step is either a turn in place or a forward movement of approximately one wheel radius (7 cm). After each step a supervisory program running on an off-board workstation recorded the rover's position and heading as computed by dead reckoning, the rover's absolute position, the state of the rover's obstacle detectors, and a time stamp. Datasets were indexed to records of the terrain layout in which they were run.

5. Results

The rover is quite slow, moving at an average speed of less than 1 cm/s. Most of the time is spent processing the data from the laser range finder. We were able to complete a total of about 100 runs over the course of a summer, of which 40 were performed in Mars-nominal terrain. The remainder were performed under a variety of other obstacle densities, including zero obstacles as a control case. (The zero-obstacle case was also used to evaluate the rover's dead-reckoning performance [18].)

The raw data consisted of a complete record of the rover's path for each run. We reduced this data by computing two performance metrics for each run: total path length and total traverse time (corrected for delays introduced by the overhead tracking system).

The reduced data for the Mars-nominal case are depicted as a scatter plot in figure 1. Each point on the plot corresponds to one run. The two axes represent the two different performance metrics. This figure illustrates our first result: path length and traverse time are poorly correlated; the correlation coefficient is 0.69. In this case the poor correlation is easily explained by the fact that the rover occasionally turns in place as part of its navigation strategy. Nevertheless, these results show that optimizing path length on the tacit assumption that other performance metrics will correlate may not be an effective strategy.

The cumulative distribution function for the reduced distance data is shown as the bold line in figure 2. (This figure shows the simulation results superimposed on the results from the real robot — see section 6. The distribution function for time looks virtually identical, but with a different scale on the y-axis.) This figure illustrates our second result: the distribution functions are not normal. In section 7 we will show that they are in fact exponential (or, at least, that an exponential distribution is a good fit) but until we do this we cannot assume any particular shape. This will complicate our analysis somewhat because most standard statistical methods assume a normal distribution.

There is to date insufficient data from the real robot in obstacle densities other than Mars nominal to allow strong quantitative conclusions to be drawn. As one would expect, performance does appear to drop off as obstacle densities increase. There is significant degradation at densities above about one obstacle per square meter, which is cause for some concern, since that is approximately the obstacle density expected at the MFEX landing site. See [18] for further discussion of these results.

6. Simulation

We constructed a simulation of the rover to serve as a reference model for making predictions about the rover's performance. Because the design of the rover is evolving we designed our simulation to be extremely flexible. Our simulator is written in Common Lisp using the Common Lisp Object System (CLOS) [24]. The code is object-oriented to an extreme. Everything in the simulator is a software object, including environments, objects in the environment, robots, sensors and actuators. Simulations are constructed by "installing" robot objects and passive objects (e.g. obstacles) into a world object. Robot objects are constructed by installing sensor and actuator objects into a "robot chassis" object. The software is designed to make it easy for users to add new sensor and actuator models, as well as new models of object interactions. The simulator naturally supports multiple-robot simulations; all that is required is to install more than one robot into the world.

The only assumption imposed by the simulator itself is that all objects with physical extent (e.g. robots, but not drive mechanisms) have that extent described by a two-dimensional polygon. (A three-dimensional model could be used if desired.) The root functionality provided by the simulator is simply efficient computation of polygon intersections. This may seem like a severe restriction, but in turns out to provide a tremendous amount of power. Furthermore, it does not preclude certain types of three-dimensional modeling. Because the interactions of objects can be defined by the user, it is possible, for example, to make a "rough terrain" object, whose physical extent is the entire world, and whose interactions with a robot are governed by a model of rough-terrain traversal provided by the user. However, we have found this to be unnecessary; a much less sophisticated model is adequate for describing the behavior of our robot in the relatively sparse obstacle fields in which we have conducted our experiments to date.

The particular model used for Rocky 3.2 was as follows:

Chassis: The robot chassis was modeled as a simple rectangle the size of the robot's footprint. At first glance this might appear to be a considerable oversimplification, but in fact the situation is not so serious. The navigation system is designed to prevent the robot's wheels from contacting any obstacles. Because the wheels are located at the perimeter of the robot's footprint it is extremely unlikely for the footprint to intersect an obstacle without at least one wheel touching that obstacle.

Drive system: The real robot uses an Ackerman drive system that also allows the three wheels on opposite sides of the chassis to be rotated in opposite directions, allowing the vehicle to turn in place. The drive mechanism can slip in loose soil. The simulated drive mechanism simply moved the robot according to the ideal kinematic model, and then added random gaussian noise to the vehicle's final position and orientation. The noise parameters were calibrated to the values measured on the real robot in the zero-obstacle experimental trials.

Rate gyro: The rate gyro included an accumulated drift error model that incorporated the observed increased error when the robot comes in contact with a rock [18].

Odometer: The odometer on the real robot measures the position of the wheels and so cannot detect when the wheels slip in loose soil. The simulation odometer mimics this behavior by recording the commanded distance traveled rather than the actual distance. The odometer data is combined with the rate gyro data to compute the simulated vehicle's dead-reckoning position.

Laser range sensor: The laser range sensor was modeled as a pair of simple proximity sensors whose outlines were the effective coverage area of the actual sensor. This simulates the behavior of the sensor after the raw range data is processed. Developing a more sophisticated model of this sensor is high on the list of potential enhancements to the system.

Despite the relatively simple nature of this simulation it produces remarkably high quality results. Qualitatively, the behavior of the simulated rover is virtually indistinguishable from that of the real one. However, this sort of gestalt assessment is precisely the sort of informal, anecdotal result that we have criticized so severely. We therefore now proceed to demonstrate formally that we have captured some of the relevant aspects of the real rover's behavior in our simulator model.

We duplicated the sandbox experiment on the simulator in two separate sets of experimental trials. In the first set of trials we did 100 runs in each of nine different terrain densities. (A random field of obstacles was generated for each run, so there were a total of 900 different obstacle fields used.) The results from these trials (excluding failures — see section 7.4) are shown in figure 2 as cumulative probability distributions. The results from the real robot are superimposed as a bold line. The Moore-model parameter ranges from 0.00015 to 0.00815 in even increments of 0.001.

In the second set of simulator trials we ran 804 runs using a Moore model parameter of 0.006. (See section 7.4 for an explanation of the apparent discrepancy between this and the value of 0.00415 used for the real experiments.) The results of these runs (again excluding failures) are shown in figure 3, superimposed with the real data and a best-fit exponential curve. By visual inspection, the fit of all three curves appears to be quite good. Selected results when failures are not excluded are shown in figure 4. As usual, the real data are superimposed as a bold line. The results do not match the real data nearly as well, indicating that the simulator's emergent failure model may be flawed.

In the next section we will formally analyze these informal observations.

7. Analysis

To draw conclusions about our data we employ statistical tests. There are a number of subtle issues in the use of statistical tests and in the interpretation of their results which are not common knowledge among mobile robotics researchers. It is therefore worthwhile to digress for a moment to discuss statistical tests in general before returning to the analysis of our data. A reader familiar with statistical methods should feel free to skip to section 7.2.

7.1 Digression: On the nature of statistical tests

Statistical analysis is related to probability theory in that both deal with phenomena that, by assumption, contain elements that are unmodelable *a priori*. However, unlike pure probability theory, which attempts to derive probabilities from first principles, statistics deals with the problem of drawing conclusions about probability distributions by examining sets of data points drawn from those distributions.

In general, a statistical analysis proceeds as follows: given a set D of samples drawn from a probability distribution P, a computation is performed on the elements of D (of which there may be any number) which yields a single result S. This result is called a statistic. (The word is also used to refer to the computation that produced S.) The familiar mean, median, mode and variance are all examples of statistics, but in general any function from sets onto scalars can serve to produce a statistic.

Statistics are themselves random variables; a statistic computed on two separate sets of samples drawn from one distribution will generally not yield the same value. However, certain statistics can be shown to have probability distributions that, under certain conditions — the so-called *null-hypothesis* conditions — are independent of the underlying distribution P which generated the datasets on which the statistic is computed. If S turns out to have a value which is very unlikely under these conditions we can confidently *reject the null hypothesis*, i.e. conclude that the conditions under which the distribution of S is known do not hold.

It is important to note that the converse reasoning is faulty. If the value of S is a likely one it does not follow that the null-hypothesis conditions are true. It is possible that the null hypothesis is false, but in a way that does not alter the distribution of S. Lack of evidence that a hypothesis H is false is not the same as evidence that H is true.

7.2 Analysis 1: exponential distribution

We wish to formally test the hypothesis that the data generated by the real robot and the simulator are drawn from exponential distributions. To do this we employ the Smirnov statistic [17] (sometimes called the Kolmogorov-Smirnov statistic), defined as:

$$KS = Max_x(|F(x) - D(x)|)$$

where F and D are cumulative probability distributions. When F and D are the same (the null hypothesis condition) the distribution of KS is independent of that of F and D. The Smirnov statistic can be used either to compare two sampling distributions (in which case the null distribution of KS depends on the number of data points in each of the two samples) or it can be used to compare a sampling distribution with an *a priori* closed-form expression. Here we will use the second method.

When we test the hypothesis that the cumulative distribution functions for our data are of the form:

$$P(x) = 1-e^{-k(x-x_0)}$$

where x is the random variable (distance or time in this case), x_0 is the smallest value of x, and k is a parameter chosen for best fit. For distance, the best fit value of k is 0.4, and for time the best fit value of k is 0.0014, yielding values of KS of 0.11 and 0.12 respectively. (The best-fit exponential curve for distance is shown in figure 3 along with the real distance data and the results of the second set of simulator runs.)

For n=40 (the number of data points) the null-distribution probabilities of the above values of KS are 0.68 and 0.60, i.e. the value of KS is expected to be at least as large as the observed values 68% and 60% of the time for sets of 40 data points drawn from the hypothesized distributions. Thus, there is no basis for rejecting the null hypothesis. (Even better fits are possible if we choose x_0 to be a value slightly less than the smallest observed values.) Note that this does not mean that the distributions *are* exponential, just that we can't distinguish any differences that there might be on the basis of the data we have.

By way of contrast, if we test the hypothesis that the distributions are normal with mean and variance equal to the sampling means and variances of the two datasets, we obtain values of KS of 0.23 for distance and 0.18 for time. The corresponding null-distribution probabilities for 40 data points are 0.025 and 0.13. Thus we can conclude with better than 95% confidence that the distribution for distance is not normal, and better than 85% confidence[1] that the distribution for time is not normal.

7.3 Analysis 2: comparison of simulated and real results

To test the second hypothesis we have three options. First, if we assume that the distributions are exponential we could employ a parametric analysis and estimation theory to derive a numerical solution (with error bounds) for the distribution functions and compare them. However, the evidence that they are in fact exponential is pretty thin, and such an assumption could lead us seriously astray. The second alternative is to use the discrete form of the Kolmogorov-Smirnov test to compare them. The third alternative is to employ a different test altogether. It turns out that for comparing two sampling distributions there are better methods available. We will use a statistic advocated by Lehmann [17], the Wilcoxon-Mann-Whitney (WMW) statistic.

The WMW statistic is computed as follows. Let D1 and D2 be sets of m and n data points drawn respectively from probability distributions P1 and P2. The data in D1 and D2 are combined and sorted. Each datum is then ranked according to its position in the sorted list; the first number in the list is assigned the rank 1, the second number the rank 2, etc. The ranks are then separated according to which distribution (D1 or D2) its corresponding datum was drawn from. The separated lists of ranks are then summed to produce two numbers, S1 and S2.

It can be shown that if P1 and P2 are the same, then S1 and S2 have normal probability distributions whose parameters are independent of P1 and P2. Instead, the parameters depend on the number of data points, m and n:

$$E(S1) = n(m+n+1)/2$$

$$E(S2) = m(m+n+1)/2$$

$$Var(S1) = Var(S2) = mn(m+n+1)/12$$

where E(X) denotes the expected value (mean) of a random variable X, and Var(x) denotes its variance. Because the variances are the same, the quantity:

$$S = S1-n(m+n+1)/2 = S2-m(m+n+1)/2$$

is sometimes used instead of S1 and S2. The variance of S is the same as that of S1 and S2, and the mean of S is, of course, zero.

[1] Usually, a confidence level of at least 90% is required for a result to be considered statistically significant. A requirement of 95% confidence is common in many fields.

Since we know that if P1 and P2 are the same then S is drawn from a normal distribution with known variance, any large deviation (relative to the variance) of S away from zero can be taken as evidence that P1 and P2 are different. If we compute:

$$p = |\Phi(S) - \Phi(-S)|$$

$$= 2\Phi(|S|)-1$$

where Φ is the error function (i.e. the integral of the normal distribution), the result is the probability that the magnitude of a sample from the null distribution of S (i.e. when P1=P2) is less than or equal to the observed value S. So, for example, if p=0.95 then there is a 95% probability that P1 and P2 are different, and only a 5% probability that P1 and P2 are the same, and that the observed value of S is due to chance.

We compared the data generated on the real rover with that generated on the second set of simulator experiments. When failures are ignored, the resulting value of p is 0.056 and 0.088 for distance and time respectively, indicating an excellent fit. (The value of p would need to be 0.90 to reject the null hypothesis.) The Smirnov probabilities for the same datasets are both greater than 99%, illustrating that the WMW test is more sensitive to differences than the Smirnov test.

7.4 Failures

Throughout the analysis we have been ignoring failures produced by the simulator. If failures are not ignored, the data produced by the simulator does not match the real robot data, indicating that the simulator's failure model is faulty. We do not have enough failure data from the real robot to properly calibrate the simulator's failure model, so we have made no attempt to correct the situation. Instead, we use this mismatch to our advantage to illustrate how our statistical procedures can detect faulty models.

Figure 4 show the distance data from the sets of three simulator runs with the best fits to the real data when failures are ignored. In this figure failures are treated as infinite distances (presumably, taking an infinitely long path to the goal is tantamount to failure), and it is clear that the match is not nearly as good as before.

Using the WMW test, the value of p for the lower curve (804 simulator runs, 175 of which were failures) is 0.98, indicating a highly statistically significant difference. The value of p for the other two curves are 0.62 and 0.64, which is not quite high enough to confidently reject the fit. (The fact that these curves have a better fit is due to the fact that the failure model was adjusted for these runs.) We would need about four times as much data from the real robot to distinguish the second two simulator distributions from the real distribution with 95% confidence, assuming that the results are reproducible. (This result was obtained by computing p for datasets consisting of multiple copies of the real data.) We are currently beginning experiments to gather this data.

Finally, although we have not yet culminated our research by verifying a statistical prediction made by our model, we have made one very interesting postdiction[2]. The best

[2] This is a postdiction rather than a prediction because the result was obtained after the experiment was completed. Experimental data collected before the formulation of a hypothesis is weaker evidence for that hypothesis than data collected afterwards. Thus, a postdiction is not as interesting as a verified prediction. However, in this case the

fit of the simulator data and the real data is obtained when the Moore model parameter in the simulator is set to approximately 0.006. However, the value of the parameter in the real experiments was 0.00415, the Mars Nominal value. This mismatch caused some consternation until it was discovered that the minimum rock diameter, D_0, in the real experiments was set to 10 cm rather than 14 as in the simulator. Moore's model is very sensitive to the value of D_0 in the 10-15 cm range, and this difference resulted in enough additional obstacles to make the overall obstacle density approximately the same as the simulator. (Actually, there were about twice as many small rocks as there should have been, but about half of those were ignored by the robot's perception system.)

8. Conclusions and Future Work

We are working towards rigorous experimental study of autonomous mobile robots. Our approach is to employ the methods of the natural sciences in our investigations. We use simulations, but we treat them as models rather than as the system under study. The value of a simulation is measured by how well it predicts the behavior of a physical system. We use statistical methods to evaluate our experimental data.

To date we have carried out only a portion of our research program. We have constructed a simulation and verified that it postdicts the behavior of a real robot in a statistically meaningful way. We have also used the simulator to generate predictions about the behavior of the rover under conditions in which it has never been tested. The final step of our research, to be completed this summer, is to test these predictions by performing a second series of experiments.

Our work offers two central contributions. We offer the first solid experimental evidence that certain performance metrics, often tacitly assumed to be well correlated, can in fact be highly uncorrelated in practice. In retrospect this is fairly obvious; nevertheless, it is a fact often ignored in the literature.

Our second contribution is the introduction of statistical rigor to the evaluation of experimental results. We have presented what is to our knowledge the first statistically significant result in the field of autonomous mobile robots, namely that the probability distributions on certain performance metrics under certain conditions are not normally distributed. We used non-parametric methods for comparing probability distribution functions. This allows us to draw quantitative conclusions about the probabilities of certain events without knowing a priori the shape of the probability distribution.

We would like to see these results independently verified by other researchers. If the shapes of the distributions on a variety of standardized tests can be conclusively established then we can transition to more powerful parametric analysis methods in comparative studies. Furthermore, if it can be established with statistical rigor that a standardized simulation is indeed an accurate model of a class of physical robots then the current ongoing debate about the value of simulation results would be settled, and the cost of conducting comparative studies of control methodologies could be dramatically reduced. However, before this can be achieved a much larger corpus of experimental data gathered under a variety of carefully controlled conditions needs to be established.

A complimentary line of research is to develop a mathematical theory to explain the observed shapes of the probability distribution functions for the performance metrics we have chosen. They appear to be exponential, indicating that rover navigation in rough terrain is a Poisson process. However, it can be argued on theoretical grounds that rover

postdiction showed that what we thought was a negative result was, in fact, a positive one by predicting a mistaken assumption in our analysis.

navigation cannot be a Poisson process because of the non-independence of obstacle encounters and the form of the termination condition. A diffusion process might be a better model. Arguments can also be made that the expected probability distributions should be two-tailed distributions (such as a chi-square), especially in denser terrains. (We have collected some preliminary data that indicate that this is not the case; even in very dense terrain the exponential distribution appears to persist.)

Finally, we also need to combine the present analysis with a similar probabilistic analysis of the performance of the robot's lookahead sensor. Measuring the performance of the lookahead sensor is currently in progress.

Acknowledgments

Reid Harrison originated and developed the support software for the virtual sandbox and performed the experimental trials on Rocky 3.2. Todd Litwin, Larry Matthies, Brian Wilcox and Wayne Zimmerman constructed the sandbox and overhead tracking system. This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Clint Bidlack, et al., "Visual Robot Navigation using Flat Earth Obstacle Projection," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[2] Johann Borenstein, "The CLAPPER: A Dual-Drive Mobile Robot with Internal Correction of Dead-Reckoning Errors," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[3] L. Feng, Y. Koren, and J. Borenstein, "A Model-Reference Adaptive Motion Controller for a Differential-Drive Robot," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[4] Erann Gat, "On the Role of Theory in the Study of Autonomous Mobile Robots," AAAI Fall Symposium on Applications of AI Theory to Real World Autonomous Mobile Robots, 1992.

[5] Erann Gat, "Robot Navigation by Conditional Sequencing," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[6] Erann Gat, et al., "Behavior Control for Robotic Exploration of Planetary Surfaces," *IEEE Transactions on Robotics and Automation,* August 1994.

[7] Steven G. Goodridge and Ren C. Luo, "Fuzzy Behavior Fusion for Reactive Control of an Autonomous Mobile Robot: MARGE," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[8] Chris Gourley and Mohan Trivedi, "Sensor-Based Obstacle Avoidance and Mapping for Fast Mobile Robots," *Proceedings of the International Conference on Robotics and Automation,* San Diego, 1994.

[9] Ian Horswill and Masaki Yamamoto. "A $1000 Active Stereo Vision System," in Proceedings of the IAPR/IEEE Workshop on Visual Behaviors, W. Martin, ed. Seattle, WA. IEEE Press, 1994.

[10] Y. Ishida, et al., "Functional Complement by Cooperation of Multiple Autonomous Robots," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[11] Hiroshi Ishiguro, et al., "A Strategy for Acquiring an Environmental Model with Panoramic Sensing by a Mobile Robot," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[12] Simon Lacroix, et al., "Autonomous Navigation in Outdoor Environment: Adaptive Approach and Experiment," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[13] James D. Lane and Robert H. King, "Computer-Assisted Guidance of an Underground Mine Truck," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[14] D. Langer, J.K. Rosenblatt, and M. Herbert, "An Integrated System for Autonomous Off-Road Navigation," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[15] Jean-Claude Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[16] Xavier Lebegue and J.K. Aggarwal, "Generation of Architectural CAD Models Using a Mobile Robot," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[17] E. L. Lehmann, *Nonparametrics: Statistical Methods Based on Ranks*, Holden-Day, 1975.

[18] Larry Matthies, et al., "Mars Microrover Navigation: Performance Evaluation and Enhancement," submitted to IROS95.

[19] H.J. Moore and B.M. Jakosky, "Viking Landing Sites, Remote Sensing Observations, and Physical Properties of Mars Surface Materials," *Icarus*, 81:164-184, 1989.

[20] Fawzi Nashashibi, et al., "3-D Autonomous Navigation in a Natural Environment," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[21] Igor Paromtchik and Ulrich Rembold, "A Practical Approach to Motion Generation and Control for an Omnidirectional Mobile Robot," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[22] Bert Schiele and James L. Crowley, "A Comparison of Position Estimation Techniques Using Occupancy Grids," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[23] Barbara Siemiatkowska, "A Highly Parallel Method for Mapping and Navigation of an Autonomous Mobile Robot," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[24] Guy L. Steele Jr., *Common Lisp: The Language*, Second Edition, Digital Press, 1990.

[25] Alexander Timcenko and Peter Allen, "Probability-Driven Motion Planning for Mobile Robots," *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

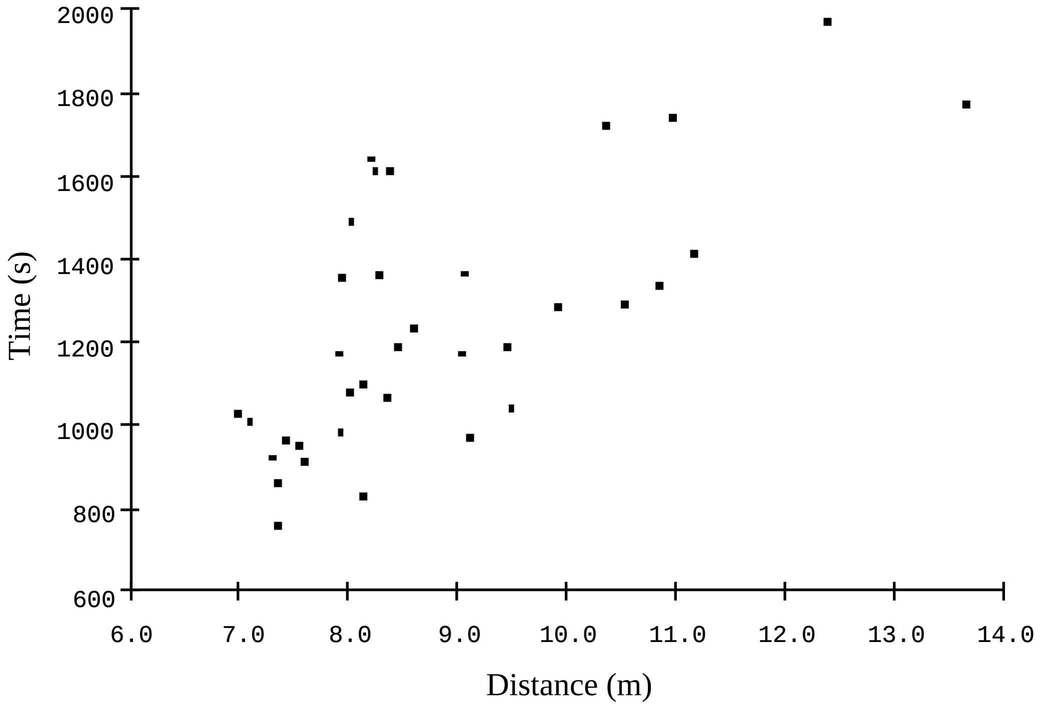

Figure 1: Scatter plot of elapsed time and distance travelled for forty runs of the real robot in Mars-nominal terrain.

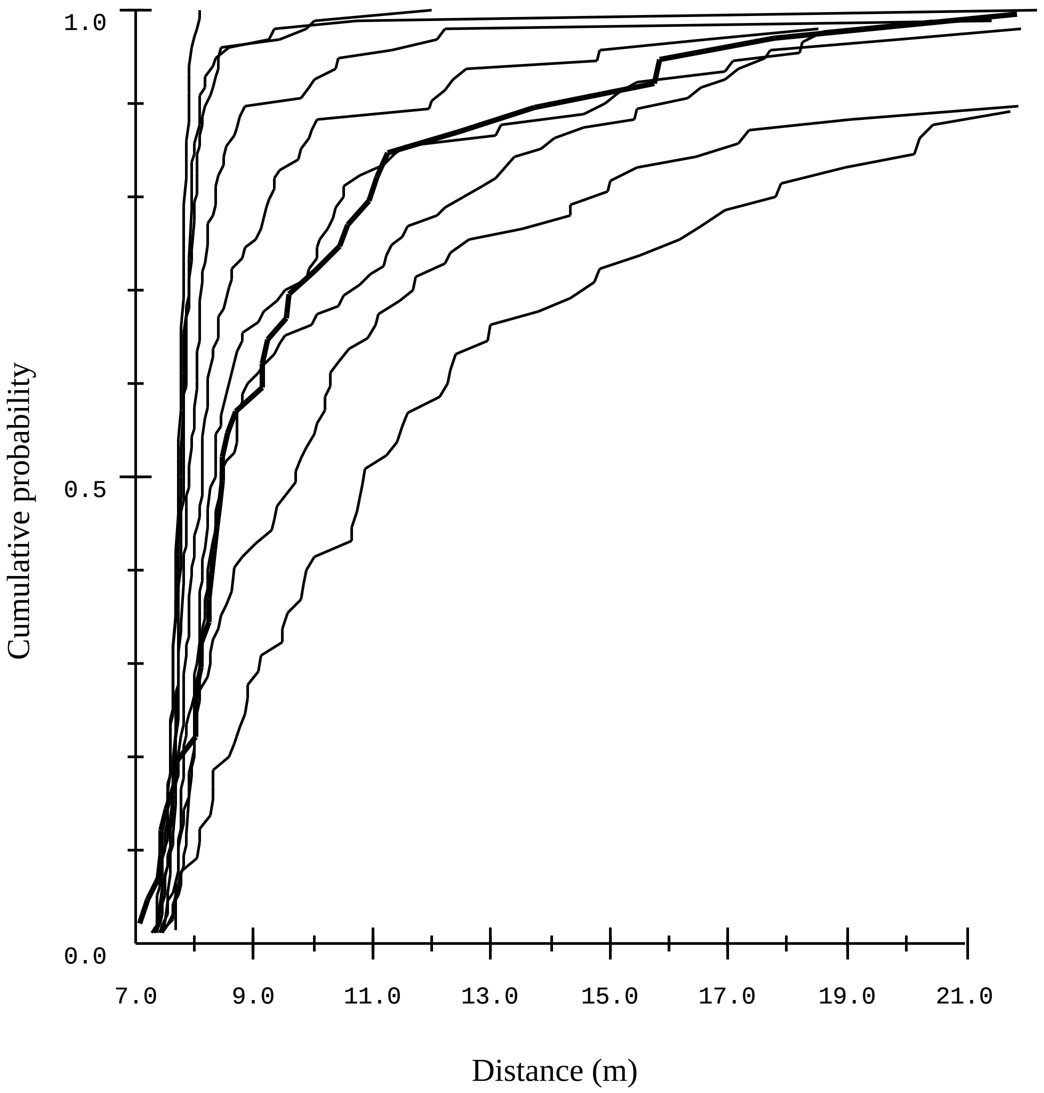

Figure 2: Cumulative probability distributions of the distance metric for a goal 7.6 meters from the starting location at various obstacle densities, superimposed on data from the real robot (bold line). The curve furthest to the left is for a Moore model parameter of 0.00015, and each successive curve increments this value by 0.001.

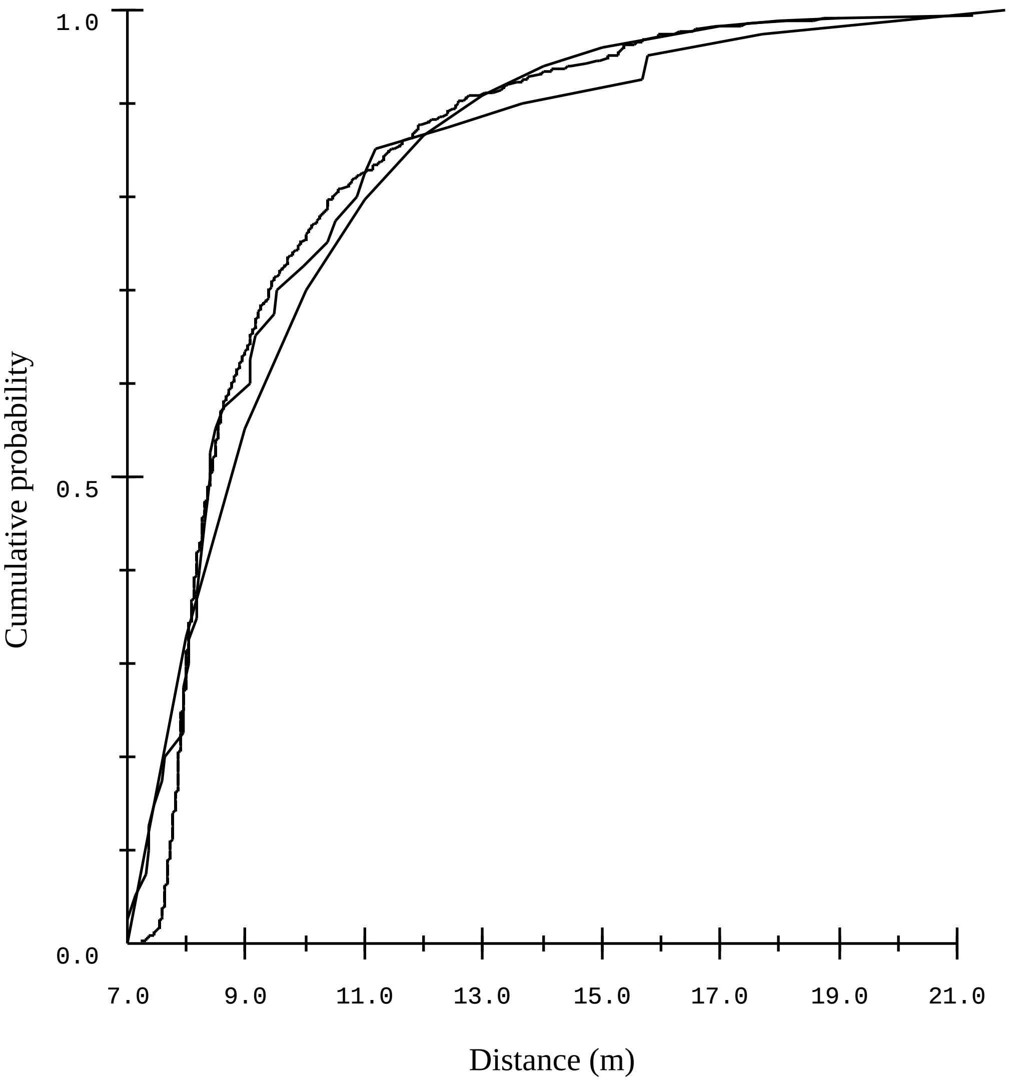

Figure 3: Comparisons of three cumulative probability functions for distance: 40 data points from the real rover, 629 data points from the simulator, and an exponential curve.

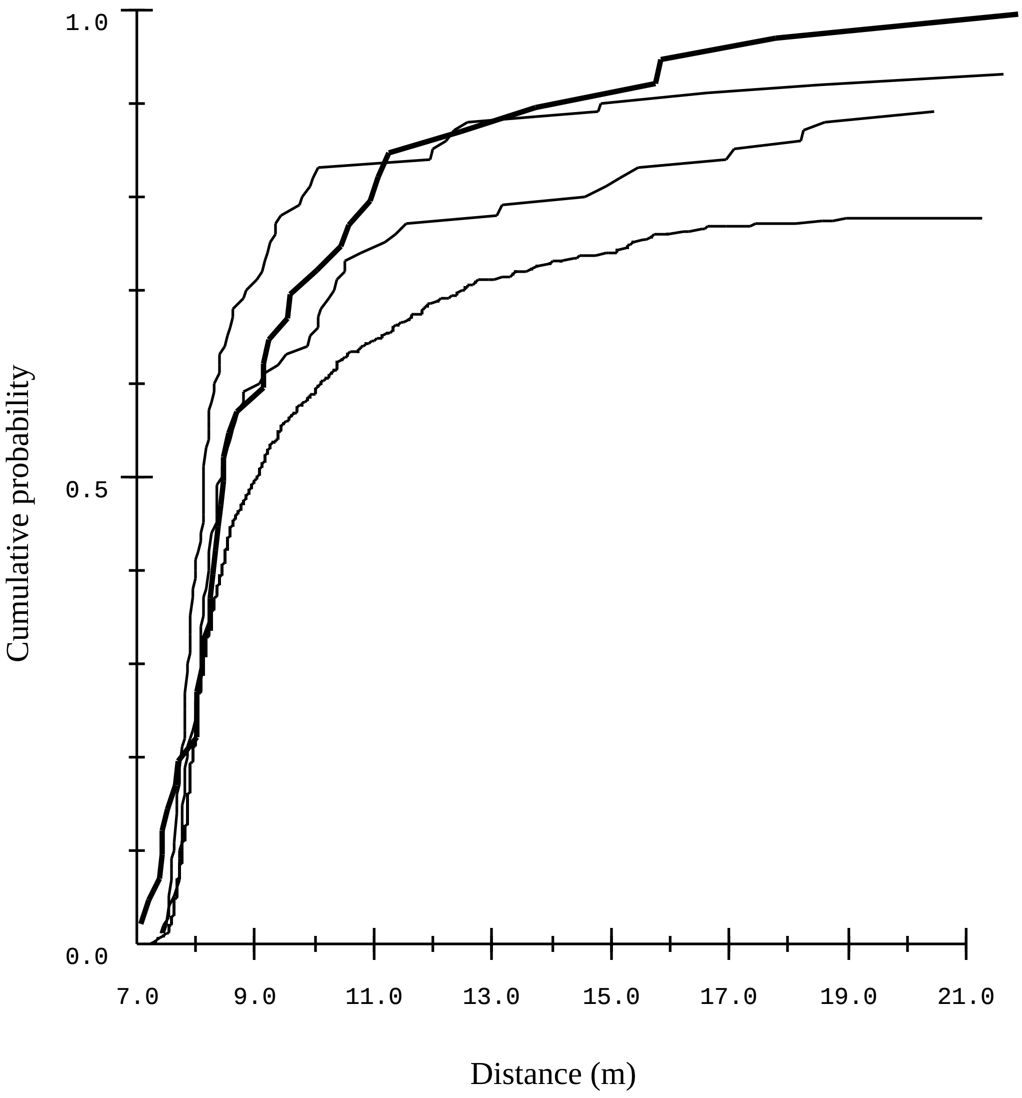

Figure 4: Simulator results when failures are not ignored for Moore parameters of 0.00415, 0.00515 and 0.00615, superimposed on the data from the real robot (bold line).